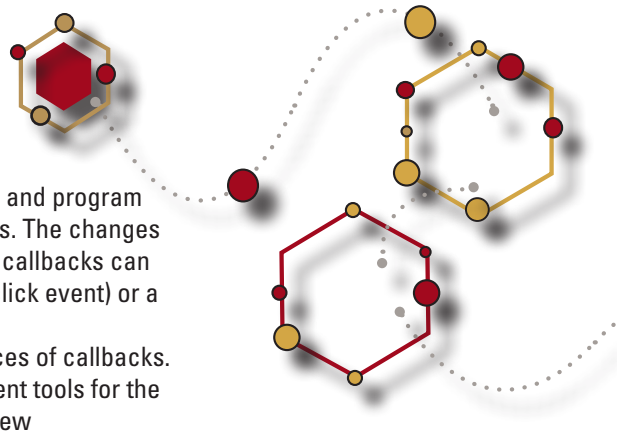
Ph.D. Preliminary Oral Defense

Friday, April 27, 2018
223 Atanasoff Hall at 1:00 p.m.

Danilo Dominguez Perez

Modeling the Control Flow of Event-Driven, Framework-Based Mobile Applications



This dissertation explores the design and implementation of new representations and program analyses for event-driven, framework-based mobile applications, specifically Android apps. The changes of control flow Android apps are mostly handled by the framework using callbacks. These callbacks can be executed synchronously and asynchronously when an external event happens (e.g. a click event) or a framework call is made.

In framework-based systems, method calls to the framework can invoke sequences of callbacks. With the high overhead introduced by libraries such as the Android framework, most current tools for the analysis of Android apps have opted to skip the analysis of these libraries. We present a new specification called Predicate Callback Summary (PCS) to summarize how library or API methods invoke callbacks. PCSs enable inter-procedural analysis for Android apps and help developers understand how their code (callback methods) is executed in the framework. We show that our static analysis techniques to summarize PCSs have positive results in terms of accuracy, considering the complexity of the millions of lines of code in the Android framework.

To integrate event-driven control flow behavior with control behavior generated from calls to the framework, without the overhead of analyzing the whole framework, we designed a novel program representation, namely Callback Control Flow Automata (CCFA). The design of CCFA is based on the Extended Finite State Machine (EFSM) model, which extends the Finite State Machine (FSM) by labeling transitions using information such as guards. In a CCFA a state represents whether the execution path enters or exits a callback. The transition from one state to another represents the transfer of control flow between callbacks. We show how CCFAs can be used to detect information flow leaks and how it can be used to implement interprocedural static analysis for Android apps.

We propose multiple projects that leverage CCFAs as the underlying framework. First, we consider that CCFAs can be useful to design new test coverage criteria given that they can serve as a gray box model for the generation of tests. We will leverage a guided model-based testing framework and use the coverage of our model to guide the generation of new tests. Second, we propose a fault explanation technique in which we develop novel demand-driven static analyses to generate a sequence of events or callbacks that trigger the bug. Lastly, we propose a work to extend CCFAs in order to detect a different kind of bugs such as race conditions.

Major Professor:
Wei Le

Committee Members:
Samik Basu

Robyn Lutz

Hridesh Rajan

Wensheng Zhang

.

IOWA STATE UNIVERSITY

Department of Computer Science

www.cs.iastate.edu